UDC: 004.93 https://doi.org/10.15407/jai2023.01.066

THE METHOD OF THE REAL-TIME HUMAN DETECTION AND TRACKING

L. Rusakova[1], **N. Shapoval**[2]
[1,2]Igor Sikorsky Kyiv Polytechnic Institute, Ukraine
37, Peremohy ave., Kyiv, 03056
rusakova.larisa@lll.kpi.ua
shovgun@gmail.com
[1]https://orcid.org/0000-0001-7849-2015

Abstract. Today, data collected from video surveillance systems require processing. Video content analysis (VCA) or video analytics (VA) has found applications in security systems, retail, the automotive industry, smart home technologies, etc. The tasks of video analytics include the detection and tracking of objects. Usually, video analytics systems are specialized software for analyzing video data collected from webcams and intelligently assessing the situation. There are also separate video surveillance cameras with built-in video analytics functions. Software approaches to human detection and tracking are quite diverse, special applications and web applications or separate software modules are created. This work presents an approach to solving these problems using convolutional neural networks. The aim of the study is to increase the efficiency of human detection and tracking in video sequences. For this purpose, an overview of existing methods of detecting objects in images was conducted, in particular: the Viola-Jones algorithm, the histogram of oriented gradients. The choice of convolutional neural networks for solving the subtask of object detection is substantiated. The working principles, advantages and disadvantages of Faster R-CNN, YOLO, SSD and RetinaNet networks are considered. Their comparative analysis was carried out according to the indicators of speed and accuracy of recognition on the HABBOF dataset. A hybrid neural network for human detection and tracking has been developed: a convolutional neural network of the YOLO type has been improved. The created network achieved an accuracy of 39.2% at 43 frames per second. Experiments were carried out with the created network in order to evaluate the operation of the network in various conditions. It has been established that it works well in poor lighting conditions, but the issue of detecting small objects remains open.

Keywords: object detection, object tracking, convolutional neural network, SSD, RetinaNet, YOLO, HOG.

Introduction

Today, video surveillance systems are used in almost all spheres of human activity. Video analytics systems provide the possibility of automatic video analysis to detect and determine temporal and spatial events [1]. These include human recognition and tracking systems used in automated driving to detect pedestrians on roads, in stores to count numbers and analyze customer behavior, at public transport stops, and in airports to analyze passenger traffic.

Formulation of the problem

Software approaches to human detection and tracking are quite diverse, special applications and web applications or separate software modules are created. Recently, neural networks have become widespread. But with this approach, there may be problems with the detection of small or incomplete objects, resistance to noisy or poorly lit images. Network training takes a lot of time, as does the detection process itself. Some networks process video at less than 10 frames per second, which is not acceptable for real-time applications. Therefore, there is a need to create an effective human detection and tracking algorithm that will allow much more accurate and faster processing of video frames.

Analysis of recent research and publications

Today, there are many different neural networks for finding objects in images or video frames. All of them are based on two approaches - one- and two-stage detection of objects. Two-stage detectors first determine the regions (areas) in which objects can be located, and then search for them in these regions. Such networks have high recognition accuracy. And single-stage detectors, which simultaneously perform localization and classification of objects in all parts of the image in one pass, demonstrate a high speed

of prediction (conclusion).

Two-stage detectors are the R-CNN network (Region-based Convolutional Neural Networks) and its derivatives (Fast R-CNN, Faster R-CNN, Mask R-CNN, Mesh R-CNN), as well as RepPoints [13], which does not use anchors. Single-stage detectors that use bindings are YOLO, SSD, and RetinaNet. Among those that do not use anchors, the CenterNet, CornerNet and FCOS networks stand out.

The aim of the study

The purpose of the study is to review the advantages and disadvantages of modern approaches to object recognition in a video stream, to compare the speed and accuracy of Faster R-CNN, YOLO, SSD, RetinaNet neural networks on the HABBOF dataset, to improve the efficiency of human detection and tracking in video sequences by creating a hybrid network: improvement of the convolutional neural network of the YOLO type.

Presenting main material

There are many different methods for detecting objects in images. Video is more complex than images because it has another dimension – time. It can be represented as a series (set) of images. And for its processing, it is enough to scroll through all the frames in the video file, apply the appropriate object recognition methods for each of the frames. But since applying the same algorithm every time is not efficient from the point of view of calculations and does not guarantee that a specific selected object will be found in each of the frames (not all detectors are resistant to different poses and angles), tracking algorithms are additionally used objects.

The Viola-Jones method [2] has achieved significant success in object detection, but it learns more slowly compared to other existing methods because it uses a significant number of features. Histogram of Oriented Gradients (Histogram of Oriented Gradients, HOG) works with local cells [3]. Therefore, it is invariant to geometric and photometric transformations of the object. But still, traditional methods of object detection are better suited for tasks where the set of training examples is small. And for working with video sequences, it is better to use convolutional neural networks.

The most important feature of neural networks, which allows you to successfully use them in various tasks (clustering, classification, pattern recognition, forecasting, etc.), is the parallel processing of information by all links, which allows you to significantly speed up the process of processing information. Another, no less important property is the ability to learn. Convolutional neural networks are effectively used in the task of object recognition in real time due to the ability to generalize the accumulated knowledge and take into account information about the ratio of image parts to each other.

Convolutional networks used for object detection are based on two approaches - one- and two-stage object detection. Two-stage detectors first determine the regions (areas) in which objects can be located, and then search for them in these regions.

Fast R-CNN shows higher accuracy and lower image processing time than R-CNN, since not all proposition regions are fed to the convolution layer. Nevertheless, an improvement of Fast R-CNN was the Faster R-CNN network, in which, instead of the resource-consuming selective search for proposition regions, a new method of object localization was implemented - RPN (Region Proporsal Network) [17]. At the heart of this approach is a system of anchors (anchor boxes, which are a combination of the center of the sliding window, scale and aspect ratio).

A feature map is formed for the image fed to the network input, which is further processed by the RPN layer. The sliding window passes through the feature map. The center of the sliding window is related to the center of the anchor. The IoU metric (Intersection over Union, also known as Jaccard index or Jaccard similarity) is used: a measure that determines the degree of intersection of regions (in the case of Faster R CNN - anchors) and is used to evaluate the similarity between two objects. Next, the feature map together with the obtained objects are transferred to the RoI layer with further classification, as well as with the determination of the displacement of the

location of potential objects. The loss function used in Faster R-CNN combines the classification loss and bounding box regression. The Faster R CNN model, in general, does a worse job of localizing objects, but is faster than Fast R-CNN. Its modification for solving the problem of image segmentation is the Mask R-CNN network [18].

The R-CNN model and similar ones achieve significant accuracy, but are very slow and only Faster R-CNN can be applied to work in real time. Such networks do not take into account the complete information about the image.

Single-stage detectors are much faster, although less accurate, than R-CNNs. The first version of YOLO (You Look Only Once) was presented in 2016. Current versions process images at up to 30 fps and have an mAP of 57.9% on the COCO (Microsoft Common Objects in Context) dataset. A key feature of YOLO is that it treats object detection as a regression problem of spatially separated bounding boxes and associated class probabilities. The input image is completely passed through the neural network in one go and objects are immediately detected.

YOLO's network architecture is inspired by the GoogLeNet model for image classification. It has 24 convolutional layers and then 2 fully connected layers. The YOLO algorithm is as follows: the image is divided into a grid of cells, which are a kind of anchors to which several bounding frames are attached. For each of them, the network displays the values of five parameters (Fig. 1): coordinates (x, y) of the center of the frame relative to the borders of the cell (cx, cy); width and height (w, h) of the frame relative to the entire image; confidence indicator (probability that the given cell contains an object).

Bounding boxes whose class probability exceeds some threshold value are selected to locate the object in the image. In order to get rid of duplicate predicted frames, the method of non-maximum suppression (NMS) is used: the frames with the maximum value of the confidence index are selected, the rest are ignored.

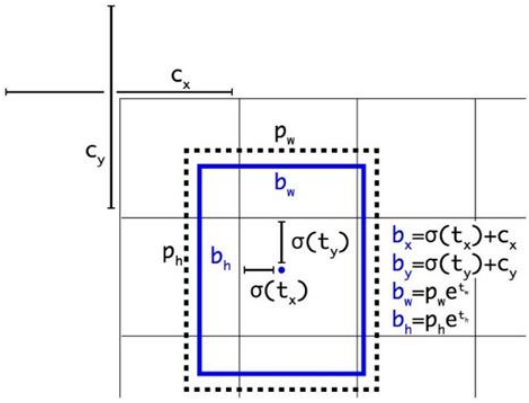


Fig. 1. Limiting frame parameters

YOLO uses the squared error between the predicted and true values to calculate a loss function that consists of:

1. classification losses (the last term; if the object is detected, it is in each cell the squared error of the conditional probabilities of each class);
2. localization losses (first two terms);
3. loss of confidence (if the object is detected, the third term is not zero, and if there is no object, the fourth term is not zero; due to the fact that most frames do not contain any objects, the problem of class imbalance arises, the coefficient λnoobj is added, which according to default is 0.5).

A limitation of YOLO is that it does not work well with small objects in the image. The next version of YOLOv2 was released a year later in 2017. The new model used the Darknet-19 network (containing 19 convolution layers and 5 max-pooling layers) as a base network. YOLOv2 increases the average accuracy using batch normalization. Another addition to the first version of YOLO was the addition of binding blocks. YOLO assumes there is one object in a cell. This becomes inconvenient when there are more objects. YOLOv2 avoids this by assuming five bounding boxes for each cell.

In 2018, the third version appeared - YOLOv3 (based on Darknet-53), which consists of 106 convolutional layers and recognizes small objects in the image with high accuracy. Allows prediction of bounding boxes at three different scales, with feature maps extracted at layers 82, 94, and 106. YOLOv3 thus compensates for YOLOv2 and YOLO's shortcomings in detecting small objects. Also, unlike its predecessors,

YOLOv3 provides three (instead of five in the previous version) bounding boxes per cell, but does so at different scales.

Another real-time object detection network somewhat similar to YOLO is SSD (Single Shot MultiBox Detector). In this case, Single Shot means that the tasks of object localization and classification are performed in one pass of the network. The SSD was one of the first single-stage detectors to achieve accuracy relatively close to two-stage detectors, while retaining the ability to operate in real time. MultiBox – a bounding box search technique, Detector: a neural network detects objects.

The main advantage of the network was a higher accuracy in the detection of small objects compared to other single-stage detectors: 77% on the Pascal VOC2007 dataset due to the detection of objects at different scales. It is more suitable for video forensics, legal detection and landmark detection.

SSD generally consists of two parts: a pre-trained network for extracting feature maps and several convolutional layers for object detection. The original SSD architecture is based on VGG 16, although a different network can be used as a basis.

VGG-16 is used as a baseline because it shows high performance in high-resolution image classification tasks. But instead of fully connected VGG layers, a set of auxiliary convolutional layers was added, which made it possible to extract objects at multiple scales and gradually reduce the size of the input data with each successive layer. To reduce the number of detected bounding frames and remove duplicates at the end, the non-maximum suppression technique is used.

Like YOLO, SSD divides the image into a grid, each grid cell responsible for detecting objects in a given area of the image. SSD has a receptive field, which allows it to detect objects at different scales, and uses default bounding boxes equivalent to bindings in Faster R-CNN. Whereas YOLO uses k-means clustering on the training dataset to determine the default bounds. Prediction for bounding boxes and confidence for different objects in the image is done using multiple feature maps of different sizes (at multiple scales). In total, SSD uses 8732 bounding frames by default. During training, they are compared in aspect ratio, location and scale to real frames. Among all rectangles, those with the largest overlap with the required bounding boxes (ie, with an IoU greater than 0.5) are selected.

A loss function is used, which is a weighted sum of localization loss and confidence loss.

There are two varieties of this model: SSD300 and SSD512. The SSD300 accepts a 300x300 image as input. It has a lower resolution, the network works faster. The SSD300 achieves 74.3% mAP at 59 FPS (frames per second) on the VOC2007 set. A 512×512, higher resolution image is fed to the SSD512 network input. This network performs more accurately, while SSD500 achieves 76.9% mAP at 22 FPS, outperforming Faster R-CNN (73.2% mAP at 7 FPS) and YOLOv1 (63.4% mAP at 45 FPS).

Another single-stage detector used for real-time object detection is RetinaNet. This network has become widely used for the analysis of aerial photographs and satellite images.

According to the authors of RetinaNet, the accuracy of single-stage detectors is lower because their training does not pay attention to the contrast between objects and the background. Therefore, a new loss function was introduced - focal loss, which focuses attention on more complex examples during network training.

RetinaNet is based on ResNet, uses a Feature Pyramid Network (FPN) for feature extraction, a network for classification of anchor blocks (with a sigmoid activation function), and a network for regression from anchor blocks to blocks with real objects.

Feature Pyramid Network (FPN) consists of three main parts: bottom-up pathway, top-down pathway and lateral connections. The ascending path is a sequence of convolutional layers with gradually decreasing dimensions. The upper layers of the convolutional network have more semantic value, but less resolution, and the lower layers have the opposite. In the original RetinaNet architecture, this function is

performed by the underlying ResNet subnet. A descending path is a sequence of layers where the feature map of the layer above gradually increases to the size of the feature map of the lower layer. Thanks to the lateral connections, the feature maps of the corresponding bottom-up and top-down layers of the pyramids are composed element by element. Due to this combination of convolutional layers, the RetinaNet architecture is scale-invariant.

As already mentioned, a new loss function is proposed in the network: focal loss (focal loss) is one of the variants of cross-entropy, which tries to solve the problem of class imbalance by assigning more weight to difficult and incorrectly classified examples (noisy background or partially overlapped objects). and a smaller one - for simple examples (background objects). The introduction of focus loss helped solve the problem of class imbalance.

Therefore, within the framework of this study, a comparative analysis of the Faster R-CNN [28], YOLO [29], SSD [30], RetinaNet [31] networks was conducted in order to assess the accuracy and speed of object detection. For this, the Human-Aligned Bounding Boxes from Overhead Fisheye cameras (HABBOF) dataset was used [26]. Developed at Boston University's Visual Information Processing (VIP) Lab and published in September 2019. The dataset contains 4 videos recorded by fisheye cameras in two different rooms (a computer lab and a small conference room) and corresponding annotations with a total of 5837 frames.

Due to the limited computational resources, the transfer learning approach was chosen for network training. Video Meeting1 was selected for training models from the dataset, and Meeting2 for testing. Pre-trained models are loaded, the necessary parameters are configured. All models were trained for 30 epochs. Network performance was compared using the following metrics:

1. Average accuracy (mAP): an indicator used to compare models and measure the accuracy of detectors; its components are precision and recall metrics [27]; accuracy is determined by the ratio of true positives (TP) to the total number of predicted positive results (the sum of true positives and false positives (FP)): $precision = \frac{TP}{TP + FP}$; completeness is the ratio of the number of true positive (TP) to the sum of true positive and false negative (FN) results: $recall = \frac{TP}{TP + FN}$; shows how many bounding boxes from the entire set were detected correctly.

2. Number of frames per second (FPS): the amount of time spent on processing one frame [27]; this metric is provided to evaluate the speed of models during inference (prediction) and can be calculated as the ratio of a unit to the network inference time (inference time); for real-time operation of the model, a value above 24 FPS is desirable.

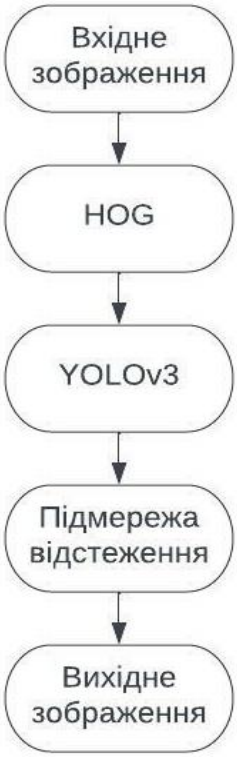
The results of the comparative analysis are shown in Table 1.

Table 1. Comparison of networks on the dataset HABBOF

Network	Basis	mAP, %	recall, %	fps
Faster R-CNN	Res Net50	51,3	31,7	9
YOLO v3	Dark Net-53	37,5	26,4	39
SSD300	VGG16	39,7	27,6	36
Retina Net	ResNet 50	41,2	33,1	28

In general, the Faster R-CNN network detects objects quite accurately (51.3% on HABBOF), but it should not be used for working with video (only 9 frames per second). SSD and RetinaNet can be applied when object detection accuracy is important. Although they are inferior to YOLO speed. At the same time, SSD detects objects at different scales, and RetinaNet has a high completeness value (33.1%), which indicates the correct classification of objects. YOLO is significantly faster (39 frames per second) and suitable for real-time operation, but shows a relatively low accuracy (37.5%). Detects more localization errors than Faster R-CNN and has difficulty detecting small and clustered objects. Therefore, during the research, it was decided to improve it.

The implemented hybrid neural network consists of the YOLOv3 network as a base for solving the object detection subtask. Since building such a model "from scratch" requires a lot of computing resources, it was decided to load a model pre-trained on the COCO dataset [33]. It is combined with the HOG descriptor and added subnet for tracking. The structure of the network is presented in Fig. 1.



Input image
HOG
YOLOv3
SubNetwork
Output image

Fig. 1. The structure of the hybrid network

The HOG descriptor extracts features from the input image and feeds them to the input of the YOLO network for object detection. Further, based on the information about the location of the central point of the object on each of the frames, the tracking of this object is implemented. For this, a PntTracking subnetwork was created, which finds consecutive pairs of nearest neighbors using two-way mapping of point descriptors.

The training of the network lasted 1 hour, which is due to the combined architecture of the network. The finished network achieved an accuracy of 39.2% at 43 frames per second.

Thus, it was found that using the HOG descriptor on this data set gave a 1.7% accuracy gain and four frames per second faster operation than using the regular YOLOv3 network.

In the future, several experiments were conducted with the proposed neural network in order to evaluate the effectiveness of its operation in various conditions. First, the video from the conference hall in the office premises was fed to the network input. A frame with the results of detection and tracking is presented in Fig. 2.

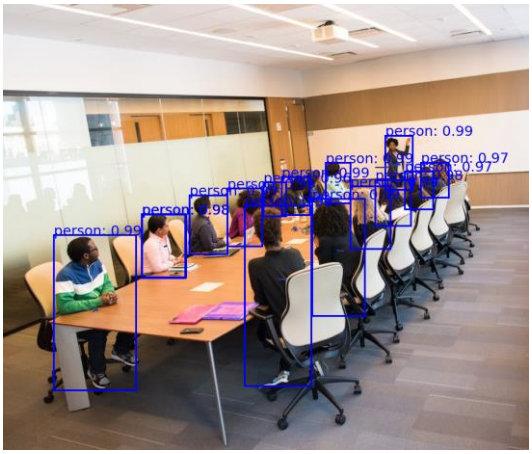


Fig. 2. Network performance results

As expected, the network quite accurately detected and tracked the people present in the video clip.

The second test was network operation in poor lighting conditions. A video fragment with pedestrians, taken in the evening, was submitted to the entrance. A frame of the resulting video is shown in Fig. 3.

The network coped with the task quite accurately. She was even able to detect people who were far from the camera and almost merged with the background.

Finally, the video from the store's surveillance cameras was submitted to the network entrance. There is a large number of small objects. Some people overlap. An example of a network result frame is presented in Fig. 4.

The network performed relatively well even with overlapping objects. However, the problem of detecting small objects still remains unsolved.

Fig. 3. Results of network operation in poor lighting conditions

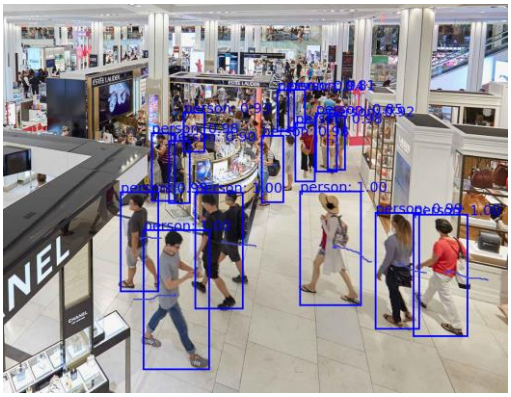


Fig. 4. Network performance results

Conclusions

Thus, in this work, the use of convolutional neural networks in the task of human detection and tracking in real time was investigated.

The obtained results are as follows. The methods of detecting and tracking objects in images were considered, in particular: the Viola-Jones algorithm, histograms of oriented gradients, neural networks, detection-based tracking. A comparative analysis of the speed and accuracy of neural networks for object detection Faster R-CNN, YOLO, SSD and RetinaNet was carried out.

The YOLO network demonstrated the greatest balance between speed and accuracy. Therefore, in the course of the study, a hybrid neural network was implemented on its basis. The combination of YOLO with the HOG descriptor made it possible to increase the accuracy and speed of detecting and tracking a person in a video stream. In addition, the network showed good results even in low light. The issue of detecting small objects remains open.

References

1. Introduction to Video Analytics. URL: https://www.eetimes.com/Introduction-to-video-analytics/ (date of application: 06.06.22).
2. Viola P., Jones M. (2001) Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition Conference (CVPR).
3. Dalal N., Triggs B. (2005) Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition Conference (CVPR).
4. Rudenko O. H., Bodyanskyi E. V. (2006) Artificial Neural Networks. Kharkiv: SMIT Company.
5. Activation functions in neural networks. URL: https://neurohive.io/ru/osnovy-data-science/activation-functions/ (date of application: 06.06.22).
6. Hubel D., Wiesel T. (1959) Receptive fields of single neurones in the cat's striate cortex. J. Physiol. 148 (3): 574-91.
7. Aston Z., Zachary C. L., Li M., Alexander J. S. (2022) Dive into Deep Learning. Release 1.0.0 alpha0.
8. Krizhevsky A., Sutskever I., Hinton G. (2012) Imagenet classification with deep convolutional neural networks. In: Conference on Neural Information Processing Systems (NIPS).
9. Szegedy C., Liu W., Jia Y. et al. (2015) Going Deeper with Convolutions. In: Conference on Computer Vision and Pattern Recognition (CVPR).
10. Simonyan K., Zisserman A. (2015) Very deep convolutional networks for large-scale image recognition. In: Conference on Neural Information Processing Systems (NIPS).
11. He K., Zhang X., Ren S., Sun J. (2016) Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR).
12. Hu J., Shen L., Sun G. (2018) Squeeze-and-Excitation Networks. In: Conference on Computer Vision and Pattern Recognition (CVPR).
13. Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, Stephen Lin. (2019) RepPoints: Point Set Representation for Object Detection. In: International Conference on Computer Vision (ICCV).
14. Girshick R., Donahue J., Darrell T., Malik J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR).
15. Girshick R. (2015) Fast R-CNN. In: International Conference on Computer Vision (ICCV).
16. Ren S., He K., Girshick R., Sun J. (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European Conference on Computer Vision (ECCV).
17. Ren S., He K., Girshick R., Sun J. (2015) Faster R CNN: Towards real-time object detection

with region proposal networks. In: Neural Information Processing Systems (NIPS).

18. He K., Gkioxari G., Dollar P., Girshick R. (2017) Mask R-CNN. In: International Conference on Computer Vision (ICCV).

19. Redmon J., Divvala S., Girshick R., Farhadi A. (2016) You only look once: Unified, real time object detection. In: Conference on Computer Vision and Pattern Recognition (CVPR).

20. Redmon J., Farhadi A. (2017) YOLO9000: better, faster, stronger. In: Conference on Computer Vision and Pattern Recognition (CVPR).

21. Redmon J., Farhadi A. (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

22. Bochkovskiy A., Wang C. Y., Liao H. Y. M. (2020) Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

23. Bochkovskiy A., Wang C. Y., Liao H. Y. M. (2021) Scaled-YOLOv4: Scaling Cross Stage Partial Network. In: Computer Vision and Pattern Recognition (CVPR).

24. Liu W. et. al. (2016) SSD: Single shot multibox detector. In European conference on computer vision.

25. Lin T. Y. et. al. (2017) Focal Loss for Dense Object Detection. In: International Conference on Computer Vision (ICCV).

26. Li S., Tezcan M. O., Ishwar P., Konrad J. (2019) Supervised people counting using an overhead fisheye camera. In: International Conference on Advanced Visual and Signal-Based Surveillance (AVSS).

27. Vavassori L. (2019) SSC: Single-Shot Multiscale Counter: Counting Generic Objects in Images.

28. Faster R-CNN model with a ResNet 50 FPN backbone.
URL: https://pytorch.org/vision/stable/models/generated/torchvision.models.detection.fasterrcnn_resnet50_fpn.html.

29. PyTorch YOLOv3.
URL: https://github.com/roboflow-ai/yolov3.

30. SSD: Single-Shot MultiBox Detector implementation in Keras.
URL: https://github.com/pierluigiferrari/ssd_keras.

31. RetinaNet model with a ResNet-50-FPN backbone.
URL: https://pytorch.org/vision/main/models/generated/torchvision.models.detection.retinanet_resnet50_fpn.html.

32. Rating of programming languages 2022.
URL:https://dou.ua/lenta/articles/language-rating-2022/ (date of application: 20.10.22).

33. YOLO: Real-Time Object Detection.
URL: https://pjreddie.com/darknet/yolo/.

The article has been sent to the editors 24.10.22
After processing 10.11.22